

✓ Exercise 1: Python Basics

Name - Anish Rao

Student Number - 20066423

Group - C

[Colab Notebook Link](#)

1. Write a program that prints "Hello, World!" to the console.

```
print("Hello, World!")
```

```
➤ Hello, World!
```

2. Calculate the area of a rectangle given its length and width.

```
def rectangle_area(length, width):  
    return length * width  
  
length = float(input("Length of the rectangle: "))  
width = float(input("Width of the rectangle: "))  
area = rectangle_area(length, width)  
print(f"Area = {area}")
```

```
➤ Length of the rectangle: 10  
Width of the rectangle: 6  
Area = 60.0
```

3. Convert temperature from Celsius to Fahrenheit and vice versa.

```
def celsius_to_fahrenheit(celsius):  
    return (celsius * 9/5) + 32  
  
def fahrenheit_to_celsius(fahrenheit):  
    return (fahrenheit - 32) * 5/9  
  
celsius = float(input("Enter temperature in Celsius: "))  
print("Celsius to Fahrenheit:", celsius_to_fahrenheit(celsius))  
fahrenheit = float(input("Enter temperature in Fahrenheit: "))  
print("Fahrenheit to Celsius:", fahrenheit_to_celsius(fahrenheit))
```

```
➤ Enter temperature in Celsius: 25  
Celsius to Fahrenheit: 77.0  
Enter temperature in Fahrenheit: 77  
Fahrenheit to Celsius: 25.0
```

4. Create a list of numbers and print the sum of all the elements.

```
numbers = [1, 2, 3, 4, 5]  
print("Sum of list:", sum(numbers))
```

```
➤ Sum of list: 15
```

5. Write a program to check if a given number is even or odd.

```
def check_even_odd(n):  
    return "Number is Even" if n % 2 == 0 else "Number is Odd"  
  
number1 = int(input("Enter a number: "))  
print(check_even_odd(number1))  
number2 = int(input("Enter a number: "))  
print(check_even_odd(number2))
```

```
➤ Enter a number: 5  
Number is Odd  
Enter a number: 6  
Number is Even
```

6. Generate a random number between 1 and 100 and ask the user to guess it.

```
import random
def guess_number():
    target = random.randint(1, 100)
    guess = int(input("Guess a number between 1-100: "))
    if guess == target:
        print("Correct")
    else:
        print(f"Wrong, correct number was: {target}")
```

```
guess_number()
```

```
➞ Guess a number between 1-100: 36
Wrong, correct number was: 15
```

7. Write a function to check if a given string is a palindrome.

```
def is_palindrome(s):
    return s == s[::-1]

print(is_palindrome("racecar"))
print(is_palindrome("hello"))
```

```
➞ True
False
```

8. Calculate the factorial of a given number.

```
def factorial(n):
    return 1 if n == 0 else n * factorial(n-1)

print(factorial(4))
print(factorial(5))
```

```
➞ 24
120
```

9. Write a program to find the largest element in a list.

```
numbers = [10, 20, 30, 40, 50, 100]
largest_number = max(numbers)
print("Largest element in the list:", largest_number)
```

```
➞ Largest element in the list: 100
```

10. Create a simple calculator that can perform addition, subtraction, multiplication, and division.

```
def calculator(a, b, operation):
    if operation == '+': return a + b
    if operation == '-': return a - b
    if operation == '*': return a * b
    if operation == '/': return a / b if b != 0 else "Cannot divide by zero"
    return "Invalid operation"

operation = input("Enter operation (+, -, *, /): ")
a = float(input("Enter first number: "))
b = float(input("Enter second number: "))
print(calculator(a, b, operation))
```

```
➞ Enter operation (+, -, *, /): +
Enter first number: 6
Enter second number: 4
10.0
```

11. Write a function to find the square root of a number using Newton's method.

```
def sqrt_newton(n, iterations=10):
    x = n
    for _ in range(iterations):
        x = 0.5 * (x + n / x)
    return x

num = float(input("Enter a number: "))
print("Square root:", sqrt_newton(num))
```

```
↵ Enter a number: 4
Square root: 2.0
```

12. Reverse a given list without using the reverse() method.

```
def reverse_list(lst):
    return lst[::-1]
print("Original list:", numbers)
print("Reversed list:", reverse_list(numbers))
```

```
↵ Original list: [10, 20, 30, 40, 50, 100]
Reversed list: [100, 50, 40, 30, 20, 10]
```

13. Write a program to find all the prime numbers between 1 and 100.

```
import math
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(math.sqrt(n)) + 1):
        if n % i == 0:
            return False
    return True
```

```
primes = [n for n in range(1, 101) if is_prime(n)]
print("Prime numbers from 1 to 100:", primes)
```

```
↵ Prime numbers from 1 to 100: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89]
```

14. Calculate the Fibonacci sequence up to a given number.

```
def fibonacci(n):
    fib_seq = [0, 1]
    while fib_seq[-1] + fib_seq[-2] <= n:
        fib_seq.append(fib_seq[-1] + fib_seq[-2])
    return fib_seq
```

```
input_number = int(input("Enter a number: "))
print(fibonacci(input_number))
```

```
↵ Enter a number: 40
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

15. Write a function to count the number of vowels in a given string.

```
def count_vowels(s):
    return sum(1 for c in s.lower() if c in "aeiou")
```

```
input_string = input("Enter a string: ")
print("No. of vowels =", count_vowels(input_string))
```

```
↵ Enter a string: my name is anish
No. of vowels = 5
```

16. Check if a given year is a leap year.

```
def is_leap_year(year):
    return (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)
```

```
year = int(input("Enter a year: "))
if is_leap_year(year):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```

```
↵ Enter a year: 2025
2025 is not a leap year.
```

17. Remove duplicates from a list without using sets.

```
def remove_duplicates(lst):
    unique = []
    for num in lst:
        if num not in unique:
```

```
        unique.append(num)
    return unique
list1 = [1, 2, 2, 3, 4, 4, 5, 6, 7]
print("Original list:", list1)
print("List without duplicates:", remove_duplicates(list1))
```

```
↻ Original list: [1, 2, 2, 3, 4, 4, 5, 6, 7]
   List without duplicates: [1, 2, 3, 4, 5, 6, 7]
```

18. Write a program to sort a list of numbers using bubble sort.

```
def bubble_sort(lst):
    n = len(lst)
    for i in range(n):
        for j in range(0, n-i-1):
            if lst[j] > lst[j+1]:
                lst[j], lst[j+1] = lst[j+1], lst[j]
    return lst
```

```
list1 = [5, 3, 8, 6, 7, 2]
print("Original list:", list1)
print("Sorted list:", bubble_sort(list1))
```

```
↻ Original list: [5, 3, 8, 6, 7, 2]
   Sorted list: [2, 3, 5, 6, 7, 8]
```

19. Create a dictionary of names and ages, and print the name of the oldest person.

```
people = {"Alice": 10, "Bob": 20, "Charlie": 30}
oldest = max(people, key=people.get)
print("Oldest person:", oldest)
```

```
↻ Oldest person: Charlie
```

20. Write a program to find the least common multiple (LCM) of two numbers.

```
def lcm(a, b):
    return abs(a * b) // math.gcd(a, b)

num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
print(f"LCM of {num1} and {num2}:", lcm(num1, num2))
```

```
↻ Enter first number: 10
   Enter second number: 15
   LCM of 10 and 15: 30
```